Markov Random Fields in Image Segmentation

Zoltan Kato

University of Szeged
Hungary
Overview

- Segmentation as pixel labeling
- Probabilistic approach
  - Markov Random Field (MRF)
  - Gibbs distribution & Energy function
- Energy minimization
  - Simulated Annealing
  - Markov Chain Monte Carlo (MCMC) sampling
- Example MRF model & Demo
- Parameter estimation (EM)
- More complex models
Segmentation as a Pixel Labelling Task

1. Extract features from the input image
   - Each pixel \( s \) in the image has a feature vector
   - For the whole image, we have
   \[
   f = \{ \vec{f}_s : s \in S \}
   \]

2. Define the set of labels \( \Lambda \)
   - Each pixel \( s \) is assigned a label \( \omega_s \in \Lambda \)
   - For the whole image, we have
   \[
   \omega = \{ \omega_s , s \in S \}
   \]

   - For an \( N \times M \) image, there are \( |\Lambda|^{NM} \) possible labelings.
   - Which one is the right segmentation?
Probabilistic Approach, MAP

- Define a *probability measure* on the set of all possible labelings and select the most likely one.
- \( P(\omega \mid f) \) measures the probability of a labelling, given the observed feature \( f \).
- Our goal is to find an optimal labeling \( \hat{\omega} \) which *maximizes* \( P(\omega \mid f) \).
- This is called the *Maximum a Posteriori* (MAP) estimate:

\[
\hat{\omega}^{MAP} = \arg \max_{\omega \in \Omega} P(\omega \mid f)
\]
Bayesian Framework

By Bayes Theorem, we have

\[ P(\omega \mid f) = \frac{P(f \mid \omega)P(\omega)}{P(f)} \propto P(f \mid \omega)P(\omega) \]

\- \( P(f) \) is constant

\- We need to define \( P(\omega) \) and \( P(f \mid \omega) \) in our model
Why MRF Modelization?

- In real images, regions are often homogenous; neighboring pixels usually have similar properties (intensity, color, texture, …)
- Markov Random Field (MRF) is a probabilistic model which captures such contextual constraints
- Well studied, strong theoretical background
- Allows MCMC sampling of the (hidden) underlying structure ➔ Simulated Annealing
What is MRF?

- To give a formal definition for Markov Random Fields, we need some basic building blocks
  - Observation Field and (hidden) Labeling Field
  - Pixels and their Neighbors
  - Cliques and Clique Potentials
  - Energy function
  - Gibbs Distribution
Definition – Neighbors

- For each pixel, we can define some surrounding pixels as its neighbors.
- Example: 1st order neighbors and 2nd order neighbors
Definition – MRF

- The labeling field \( X \) can be modeled as a Markov Random Field (MRF) if
  1. For all \( \omega \in \Omega : P(X = \omega) > 0 \)
  2. For every \( s \in S \) and \( \omega \in \Omega : \)
     \[
     P(\omega_s | \omega_r, r \neq s) = P(\omega_s | \omega_r, r \in N_s)
     \]

\( N_s \) denotes the neighbors of pixel \( s \)
Hammersley-Clifford Theorem

- The Hammersley-Clifford Theorem states that a random field is a MRF if and only if $P(\omega)$ follows a Gibbs distribution.

\[
P(\omega) = \frac{1}{Z} \exp(-U(\omega)) = \frac{1}{Z} \exp(-\sum_{c \in C} V_c(\omega))
\]

- where $Z = \sum_{\omega \in \Omega} \exp(-U(\omega))$ is a normalization constant.

- This theorem provides us an easy way of defining MRF models via \textit{clique potentials}. 

Definition – Clique

- A subset $C \subseteq S$ is called a **clique** if every pair of pixels in this subset are neighbors.
- A clique containing $n$ pixels is called $n^{th}$ **order clique**, denoted by $C_n$.
- The set of cliques in an image is denoted by

$$C = C_1 \cup C_2 \cup \ldots \cup C_k$$
Definition – Clique Potential

- For each clique $c$ in the image, we can assign a value $V_c(\omega)$ which is called \textit{clique potential} of $c$, where $\omega$ is the configuration of the labeling field.

- The sum of potentials of all cliques gives us the energy $U(\omega)$ of the configuration $\omega$.

$$U(\omega) = \sum_{c \in C} V_c(\omega) = \sum_{i \in C_1} V_{C_1}(\omega_i) + \sum_{(i,j) \in C_2} V_{C_2}(\omega_i, \omega_j) + \ldots$$
Segmentation of grayscale images: A simple MRF model

- Construct a segmentation model where regions are formed by spatial clusters of pixels with similar intensity:

\[
\hat{\omega}
\]

- MRF segmentation model
- Model parameters
- Input image
- Find MAP estimate
- Segmentation

\[
\hat{\omega}
\]
MRF segmentation model

- Pixel labels (or classes) are represented by Gaussian distributions:

\[
P(f_s | \omega_s) = \frac{1}{\sqrt{2\pi \sigma_{\omega_s}^2}} \exp\left(-\frac{(f_s - \mu_{\omega_s})^2}{2\sigma_{\omega_s}^2}\right)
\]

- Clique potentials:
  - **Singleton**: proportional to the likelihood of features given \(\omega\): \(\log(P(f | \omega))\).
  - **Doubleton**: favours similar labels at neighbouring pixels – *smoothness prior*

\[
V_{c_2}(i, j) = \beta \delta(\omega_i, \omega_j) = \begin{cases} 
-\beta & \text{if } \omega_i = \omega_j \\
+\beta & \text{if } \omega_i \neq \omega_j
\end{cases}
\]

As \(\beta\) increases, regions become more homogenous
Model parameters

- **Doubleton potential** $\beta$
  - less dependent on the input ➔
    - can be fixed a priori

- **Number of labels** $|\Lambda|$
  - Problem dependent ➔
    - usually given by the user or
    - inferred from some higher level knowledge

- Each label $\lambda \in \Lambda$ is represented by a Gaussian distribution $N(\mu_\lambda, \sigma_\lambda)$:
  - estimated from the input image
Model parameters

- The class statistics (mean and variance) can be estimated via the empirical mean and variance:

\[
\forall \lambda \in \Lambda : \quad \mu_\lambda = \frac{1}{|S_\lambda|} \sum_{s \in S_\lambda} f_s ,
\]

\[
\sigma_\lambda^2 = \frac{1}{|S_\lambda|} \sum_{s \in S_\lambda} (f_s - \mu_\lambda)^2
\]

- where \( S_\lambda \) denotes the set of pixels in the training set of class \( \lambda \)
- a training set consists in a representative region selected by the user
Energy function

- Now we can define the energy function of our MRF model:

\[
U(\omega) = \sum_s \left( \log(\sqrt{2\pi\sigma_{\omega_s}}) + \frac{(f_s - \mu_{\omega_s})^2}{2\sigma_{\omega_s}^2} \right) + \sum_{s,r} \beta\delta(\omega_s, \omega_r)
\]

- Recall:

\[
P(\omega | f) = \frac{1}{Z} \exp(-U(\omega)) = \frac{1}{Z} \exp(-\sum_{c \in C} V_c(\omega))
\]

- Hence

\[
\hat{\omega}^{MAP} = \arg \max_{\omega \in \Omega} P(\omega | f) = \arg \min_{\omega \in \Omega} U(\omega)
\]
Optimization

- Problem reduced to the minimization of a **non-convex** energy function
  - Many local minima
- Gradient descent?
  - Works only if we have a *good* initial segmentation
- Simulated Annealing
  - Always works (at least in theory)
ICM (~Gradient descent) [Besag86]

1. Start at a “good” initial configuration $\omega^0$ and set $k = 0$.

2. For each configuration which differs at most in one element from the current configuration $\omega^k$ (they are denoted by $N_{\omega^k}$), compute the energy $U(\eta)$ ($\eta \in N_{\omega^k}$).

3. From the configurations in $N_{\omega^k}$, select the one which has a minimal energy:

$$\omega^{k+1} = \arg \min_{\eta \in N_{\omega^k}} U(\eta). \quad (6)$$

4. Go to Step 2 with $k = k + 1$ until convergence is obtained (for example, the energy change is less than a certain threshold).
Simulated Annealing

① Set $k = 0$ and initialize $\omega$ randomly. Choose a sufficiently high initial temperature $T = T_0$.

② Construct a trial perturbation $\eta$ from the current configuration $\omega$ such that $\eta$ differs only in one element from $\omega$.

③ (Metropolis criteria) Compute $\Delta U = U(\eta) - U(\omega)$ and accept $\eta$ if $\Delta U < 0$ else accept with probability $\exp(-\Delta U/T)$ (analogy with thermodynamics):

$$
\omega = \begin{cases} 
\eta & \text{if } \Delta U \leq 0, \\
\eta & \text{if } \Delta U > 0 \text{ and } \xi < \exp(-\Delta U/T), \\
\omega & \text{otherwise}
\end{cases}
$$

(4)

where $\xi$ is a uniform random number in $[0, 1)$.

④ Decrease the temperature: $T = T_{k+1}$ and goto Step ② with $k = k + 1$ until the system is frozen.
Temperature Schedule

\[ T_k \geq \frac{\Gamma}{\ln(k)} \]  \hspace{1cm} (8)

with

\[ \Gamma > \max_{\omega \in \Omega} U(\omega) - \min_{\omega \in \Omega} U(\omega) \]  \hspace{1cm} (9)
Temperature Schedule

- **Initial temperature**: set it to a relatively low value (~4) to ensure faster execution
  - must be high enough to allow random jumps at the beginning!
- **Schedule**: \[ T_{k+1} = c \cdot T_k, \quad k = 0, 1, 2, \ldots \]
- **Stopping criteria**:  
  - Fixed number of iterations  
  - Energy change is less than a threshold
Demo

- Download from:
  [http://www.inf.u-szeged.hu/~kato/software/](http://www.inf.u-szeged.hu/~kato/software/)
Summary

- Design your model carefully
  - Optimization is just a tool, do not expect a good segmentation from a wrong model
- What about other than graylevel features
  - Extension to color is relatively straightforward
What color features?

- RGB histogram
- CIE-L* a* b* histogram
Extract Color Feature

- We adopt the CIE-L\textsuperscript{*}u\textsuperscript{*}v\textsuperscript{*} color space because it is \textit{perceptually uniform}.
  - Color difference can be measured by Euclidean distance of two color vectors.
- We convert each pixel from RGB space to CIE-L\textsuperscript{*}u\textsuperscript{*}v\textsuperscript{*} space
  - We have 3 color feature images

![Color feature images](image-url)
Color MRF segmentation model

- Pixel labels (or classes) are represented by three-variate Gaussian distributions:

\[
P(f_s | \omega_s) = \frac{1}{\sqrt{(2\pi)^n | \Sigma_{\omega_s}|}} \exp\left(-\frac{1}{2} (\tilde{f}_s - \tilde{u}_{\omega_s})\Sigma_{\omega_s}^{-1}(\tilde{f}_s - \tilde{u}_{\omega_s})^T\right)
\]

- Clique potentials:
  - **Singleton**: proportional to the likelihood of features given \( \omega \): \( \log(P(f | \omega)) \).
  - **Doubleton**: favours similar labels at neighbouring pixels – *smoothness prior*

\[
V_{c_2}(i, j) = \beta \delta(\omega_i, \omega_j) = \begin{cases} 
-\beta & \text{if } \omega_i = \omega_j \\
+\beta & \text{if } \omega_i \neq \omega_j 
\end{cases}
\]

As \( \beta \) increases, regions become more homogenous.
Summary

- Design your model carefully
  - Optimization is just a tool, do not expect a good segmentation from a wrong model
- What about other than graylevel features
  - Extension to color is relatively straightforward
- Can we segment images without user interaction?
  - Yes, but you need to estimate model parameters automatically (EM algorithm)
Incomplete data problem

- Supervised parameter estimation
  - we are given a labelled data set to learn from
    - e.g. somebody manually assigned labels to pixels
- How to proceed without labelled data?
  - Learning from incomplete data
  - Standard solution is an iterative procedure called **Expectation-Maximization**
    - Assigns labels and estimates parameters simultaneously
    - Chicken-Egg problem
EM principles: The two steps

**E Step**: For each pixel, use parameters to compute probability distribution.

**Parameters**: \( P(\text{pixel}/\text{label})P(\text{label}) \)

**Weighted labeling**: \( P(\text{label}/\text{pixel}) \)

**M Step**: Update the estimates of parameters based on weighted (or “soft”) labeling.
The basic idea of EM

- Each of the **E** and **M** steps is straightforward assuming the other is solved
  - Knowing the label of each pixel, we can estimate the parameters
    - Similar to supervised learning (hard vs. soft labeling)
  - Knowing the parameters of the distributions, we can assign a label to each pixel
    - by Maximum Likelihood – i.e. using the singleton energies only without pairwise interactions
Parameter estimation via EM

- Basically, we will fit a mixture of Gaussian to the image histogram
  - We know the number of labels $|\Lambda| \equiv$ number of mixture components

- At each pixel, the complete data includes
  - The observed feature $f_s$
  - Hidden pixel labels $l_s$ (a vector of size $|\Lambda|$)
    - specifies the contribution of the pixel feature to each of the labels – i.e. a soft labeling
Parameter estimation via EM

- **E** step: recompute $I_s^i$ at each pixel $s$:

  $$I_s^i = P(\lambda \mid f_s) = \frac{P(f_s \mid \lambda)P(\lambda)}{\sum_{\lambda \in \Lambda} P(f_s \mid \lambda)P(\lambda)}$$

- **M** step: update Gaussian parameters for each label $\lambda$:

  $$P(\lambda) = \frac{\sum_{s \in S} P(\lambda \mid f_s)}{|S|}, \quad \mu_\lambda = \frac{\sum_{s \in S} P(\lambda \mid f_s)f_s}{\sum_{s \in S} P(\lambda \mid f_s)}$$
Summary

- Design your model carefully
  - Optimization is just a tool, do not expect a good segmentation from a wrong model
- What about other than graylevel features
  - Extension to color is relatively
- Can we segment images without user interaction?
  - Yes, but you need to estimate model parameters automatically (EM algorithm)
- What if we do not know $|\Lambda|$?
  - Fully automatic segmentation requires
    - Modeling of the parameters AND
    - a more sophisticated sampling algorithm (Reversible jump MCMC)
MRF+RJMCMC vs. JSEG

**JSEG** (Y. Deng, B.S.Manjunath: PAMI’01):

1. **color quantization**: colors are quantized to several representing classes that can be used to differentiate regions in the image.
2. **spatial segmentation**: A region growing method is then used to segment the image.
Benchmark results using the Berkeley Segmentation Dataset

JSEG

RJMCMC
Summary

- Design your model carefully
  - Optimization is just a tool, do not expect a good segmentation from a wrong model
- What about other than grayscale features
  - Extension to color is relatively
- Can we segment images without user interaction?
  - Yes, but you need to estimate model parameters automatically (EM algorithm)
- What if we do not know $|\Lambda|$?
  - Fully automatic segmentation requires
    - Modeling of the parameters AND
    - a more sophisticated sampling algorithm (Reversible jump MCMC)
- Can we segment more complex images?
  - Yes, but you need a more complex MRF model
Objectives

- Combine different segmentation cues:
  - Color & Texture [ICPR2002, ICIP2003]
  - Color & Motion [ACCV2006, ICIP2007]
  - …?

- How humans do it?
  - Multiple cues are perceived simultaneously and then they are integrated by the human visual system [Kersten et al. An. Rev. Psych. 2004]
  - Therefore different image features has to be handled in a parallel fashion.

- We attempt to develop such a model in a Markovian framework
Multi-Layer MRF Model: Neighborhood & Interactions

- $\omega$ is modeled as a MRF
  - Layered structure
  - “Soft” interaction between features
- $P(\omega \mid f)$ follows a Gibbs distribution
  - Clique potentials define the local interaction strength
- MAP $\iff$ Energy minimization ($U(\omega)$)

Hammersley-Clifford Theorem:

$$P(\omega) = \frac{\exp(-U(\omega))}{Z} = \frac{\exp(-\sum_c V_c(\omega))}{Z}$$

Model $\iff$ Definition of clique potentials
Texture Layer: MRF model

- We extract two types of texture features
  - *Gabor feature* is good at discriminating strongly ordered textures
  - *MRSAR feature* is good at discriminating weakly ordered (or random) textures
- The number of texture feature images depends on the size of the image and other parameters.
  - Most of these don’t contain useful information
  - Select feature images with high discriminating power.
- MRF model is similar to the color layer model.
Examples of Texture Features

Gabor features:

MRSAR features:
Combined Layer: Labels

- A label on the combined layer consists of a pair of color and texture/motion labels such that $\eta_s = \langle \eta^c_s, \eta^m_s \rangle$
  where $\eta^c_s \in \Lambda^c$ and $\eta^m_s \in \Lambda^n$
- The number of possible classes is $L^c \times L^m$
- The combined layer selects the most likely ones.
Combined Layer: **Singleton** potential

- Controls the number of classes:

\[ V_s(\eta_s) = R\left((10N\eta_s)^{-3} + P(L)\right) \]

- \(N\eta_s\) is the percentage of labels belonging to class \(\eta_s\)
- \(L\) is the number of classes present on the combined layer.

- Unlikely classes have a few pixels \(\Rightarrow\) they will be penalized and removed to get a lower energy

- \(P(L)\) is a log-Gaussian term:
  - Mean value is a guess about the number of classes,
  - Variance is the confidence.
Combined Layer: **Doubleton** potential

- Preferences are set in this order:
  1. Similar color and motion/texture labels
  2. Different color and motion/texture labels
  3. Similar color (resp. motion/texture) and different motion/texture (resp. color) labels

- These are contours visible only at one feature layer.

\[
\delta(\eta_s, \eta_r) = \begin{cases} 
-\alpha & \text{if } \eta_s^c = \eta_r^c, \eta_s^m = \eta_r^m \\
0 & \text{if } \eta_s^c \neq \eta_r^c, \eta_s^m \neq \eta_r^m \\
+\alpha & \text{if } \eta_s^c \neq \eta_r^c, \eta_s^m = \eta_r^m \\
& \text{or } \eta_s^c = \eta_r^c, \eta_s^m \neq \eta_r^m 
\end{cases}
\]
Inter-layer clique potential

- Five pair-wise interactions between a feature and combined layer
- Potential is proportional to the difference of the singleton potentials at the corresponding feature layer.
  - Prefers $\omega_s$ and $\eta_s$ having the same label, since they represent the labeling of the same pixel
  - Prefers $\omega_s$ and $\eta_r$ having the same label, since we expect the combined and feature layers to be homogenous
Color Textured Segmentation
Color Textured Segmentation

Table: Color Textured Segmentation

<table>
<thead>
<tr>
<th>Original Image</th>
<th>Texture Segmentation</th>
<th>Color Segmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Original Image" /></td>
<td><img src="image2.png" alt="Texture Segmentation" /></td>
<td><img src="image3.png" alt="Color Segmentation" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Multi-cue Segmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Texture Layer Result</td>
</tr>
<tr>
<td>Color Layer Result</td>
</tr>
<tr>
<td>Combined Layer Result</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Original Image</th>
<th>Texture Segmentation</th>
<th>Color Segmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image4.png" alt="Original Image" /></td>
<td><img src="image5.png" alt="Texture Segmentation" /></td>
<td><img src="image6.png" alt="Color Segmentation" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Multi-cue Segmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Texture Layer Result</td>
</tr>
<tr>
<td>Color Layer Result</td>
</tr>
<tr>
<td>Combined Layer Result</td>
</tr>
</tbody>
</table>
Color & Motion Segmentation
References

- Visit http://www.inf.u-szeged.hu/~kato/